User Manual S.USV pi solutions

Compatible with S.USV pi basic and S.USV pi advanced
Revision 1.0 | Date 10.10.2015

Table of Contents

1 Functions 3
2 Technical Specification 4
2.1 Overview 4
2.2 Performance 4
2.3 Lighting Indicators 5
3 Installation Guide 6
3.1 Hardware 6
3.2 Software 6
3.2.1 Raspbian 6
3.2.2 I2C 7
3.2.3 S.USV 8
3.2.4 RTC – Real Time Clock 9
4 Client Software 10
4.1 S.USV – Daemon 11
4.1.1 Daemon Configuration 11
4.1.2 Daemon Controlling 12
4.2 S.USV – Client 12
4.2.1 Client Options 12

1 Functions

The S.USV pi solutions is an advanced power supply additional module for Raspberry Pi, with the main focus on the uninterruptible power supply of the single-board computer.
The module also provides additional functions in order to optimize the operation of the Raspberry Pi by the user.

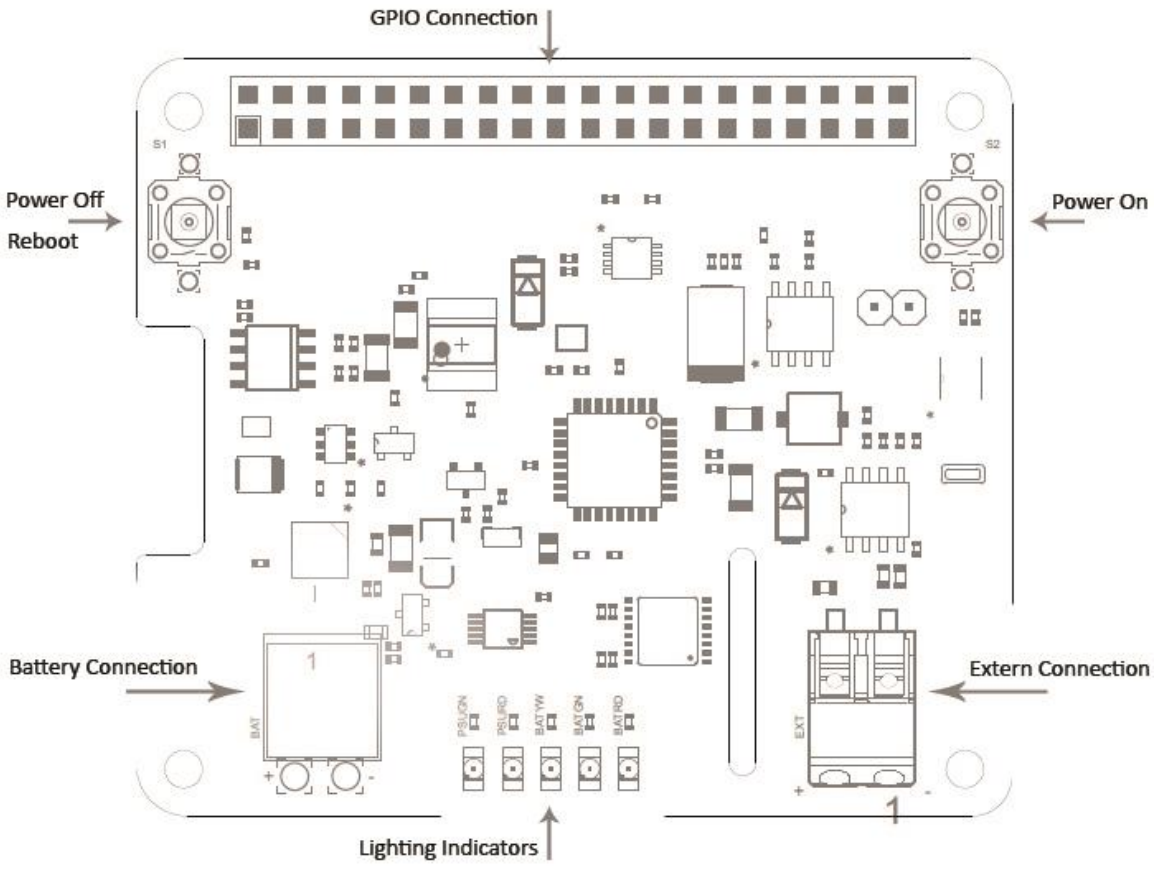
The S.USV pi solutions is a fully functional plug & play solution. The power supply occurs directly through the J8 connector on the Raspberry Pi and therefore uses a common voltage source, thus no additional cabling or power supply needed. In addition, the module is equipped with a LiPo battery. An integrated boost switching power converter covers the necessary voltage range, thereby the Raspberry Pi shut down safely in case of misconduct and prevent data loss.
The “advanced” version also provides a power input for the extended voltage range of 7 – 24 volts (solar cells, automotive applications, etc.).

- HAT compliant UPS Module
- Compatible with Pi 2 B, Pi A+, Pi B+
- Adapter solution for Pi Model A and B
- Uninterruptible power supply
- Plug & Play (Power via Raspberry Pi)
- Monitoring – System
- Intelligent software solution including mobile application
- Built-in LiPo battery (300mAh) with intelligent charging function
- Battery Management Controller
- Battery Monitoring System
- Software simulated Real Time Clock with Battery Back-Up
- Time-controlled on and off switching of the Raspberry Pi
- Raspberry Pi
- Supply Switch (Power on and off buttons / File safe shutdown)
- LED – Status display
- Bootloader for Live – Firmware updates
- Power input with extended voltage range of 7-24 volts

2 Technical Specification

	S.USV *pi basic*	S.USV *pi advanced*
Plug & Play	√	√
Power Supply	5 Volts/2000 mA	5 Volts/2000 mA
Extended Voltage Range	x	7-24 Volts/2000 mA
Interfaces	I²C	I²C
ID EEPROM	√	√
Monitoring - System	√	√
Mobile Application	√	√
LiPo - Battery	√	√
Battery Management	√	√
Battery Monitoring	√	√
Real Time Clock	√	√
Raspberry Pi Supply Switch	√	√
Compatibility	Pi 2 Model B, Pi A+, Pi B+	Pi 2 Model B, Pi A+, Pi B+
Dimensions	65x56,5 mm	65x56,5 mm

2.1 Overview



2.2 Performance

- **Battery Connection:** Connector for connecting the supplied LiPo battery.
  Battery Connector: (Würth Elektronik 620 002 113 322)
- **Extern Connection:** Connector for connecting extended voltage range (7-24V).
- **GPIO PORT:** GPIO Connection to the Raspberry Pi.
- **Power Off / Reboot:** Press for reboot or hold at least 3 seconds for shutdown.
- **Power On:** Press for booting the Raspberry Pi.
- **Lighting Indicators:** LED status indicator for the S.USV / Raspberry Pi.

2.3 Lighting Indicators

LED	Indication
PSU GREEN (Blinking)	Startup - initialization of the S.USV firmware
PSU GREEN	RPi Power Supply Unit is online (Voltage present)
PSU RED	RPi Power Supply Unit is offline (Voltage loss) – Battery Powering is online
BAT YELLOW	Charging Circuit Online – Battery is charging
BAT GREEN	Charging Circuit Online – Battery is fully charged
BAT RED	Charging Circuit Offline – Battery is missing or corrupt
BAT RED (Blinking)	Charging Circuit Offline – Remaining battery capacity in the critical area

3 Installation Guide

3.1 Hardware

Before first start-up of the S.USV *pi solutions*, please make sure that the supplied battery is properly plugged into the provided JST connector on the front side of the board.

In initial operation we recommend to fully charge the supplied battery to ensure full functionality. Furthermore, we recommend a PSU with at least 2 amps to operate the Raspberry Pi.

3.2 Software

(Note: The current software has been optimized specifically for Raspbian. In the next few weeks further versions for all common operating systems will be available.)

3.2.1 Raspbian

- To install and configure the Raspberry Pi, we recommend the Quick Start Guide of Raspberry Pi directly:

  https://www.raspberrypi.org/help/quick-start-guide/

- To install the operating system, we recommend the Image Installation Guides of Raspberry Pi:

  https://www.raspberrypi.org/documentation/installation/installing-images/

- The Image of Raspbian operating system can be found on the following page:

  https://www.raspberrypi.org/downloads/

3.2.2 I2C

The ID EEPROM contains data that identifies the board, tells the Raspberry Pi how the GPIOs need to be set up and what hardware is on the board. This allows the S.USV *pi solutions* to be automatically identified and set up by the Pi software at boot time including loading all the necessary drivers.

Users of previous models A and B please follow the manual instructions:

- The communication between the S.USV and the Raspberry Pi happens via the I2C - interface, please activate and configure this first using the following steps:

  1. First you have to install the relevant I2C-Tools to be able to see which devices are connected to your Raspberry Pi. To do this, you have to enter the following commands in the Terminal to install the i2c-tools utility:

     sudo apt-get install python-smbus

     sudo ap-get install i2c-tools

  2. Next you will need to open LXTerminal or console or SSH and enter the following command:

     sudo nano /etc/modules

     and add these two lines to the end of the file

     i2c-bcm2708
     i2c-dev

     Then save the file with **Control-X Y** <return>

  3. Depending on your Distribution, you may also have a file called ***/etc/modprobe.d/raspi-blacklist.conf***

     If you do not have this file, then there is nothing to do, however, if you have this file, you have to edit and comment out the lines below by putting a # in front of them:

     blacklist spi-bcm2708
     blacklist i2c-bcm2708

     Open an Editor on the file by typing: ***sudo nano /etc/modprobe.d/raspi-blacklist.conf*** then edit the file as described above and then save and exit the file using **Control-X Y** <return>

4. If you are running a recent Raspberry Pi (3.18 kernel or higher) you will also need to update the ***/boot/config.txt*** file. Edit it with ***sudo nano /boot/config.txt*** and add the following lines to the bottom of the file:

   dtparam=i2c1=on
   dtparam=i2c_arm=on

5. Once, the steps above are done, reboot the system by using the command ***sudo reboot***

6. Now, when you log in, you can use the following command to see all the connected devices:

   sudo i2cdetect –y 1

   This shows, that two I2C-Adresses are in use – 0x0F for the S.USV and 0x68 for the Real Time Clock located on the S.USV.
   Note, that if you are using one of the first Raspberry Pis, you will have to change the command to ***sudo i2cdetect –y 0***

3.2.3 S.USV

1. Please download the provided debian package from our download section and save it to any local memory address on your Raspberry Pi.

2. To install the Debian package on your Raspberry Pi switch to directory in which the debian package is located and perform the following commands in the command line.

   sudo tar xvfz susvd-en-1.0-all.tar.gz (to unzip the tar file)

   sudo dpkg –i susvd-en-1.0-all.deb (to install the debian package)

3. The S.USV Client and Daemon are now fully installed and ready for use.
   (The installed files are located in the following path: /opt/susvd)

4. If the installation was successful you have to start the Daemon in order for the S.USV to work correctly by using the following command (Refer to section 4 for a description of all the commands):

   sudo ./susvd –start

   The correct start of the Daemon will be shown in the console output.

5. If everything is done, you can use the full functionality of the S.USV.

3.2.4 RTC – Real Time Clock

The ID EEPROM on the board configures the RTC module automatically.

(Note: Refer to point 2 to set up the current time.)

Users of previous models A and B please follow the manual instructions:

- Please verify that the chip module is seen by running ***sudo i2cdetect –y 1*** (i2cdetect –y 0 for old Rev. 1 Pi) at the command line. The DS1307 Real Time Clock should be located at the I2C address ID #68.

1. Load the RTC module by running ***sudo modprobe rtc-ds1307*** in the kernel. The following commands should be running as super user.

   For a Rev. 2 Pi or later:

   echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device

   For the older Rev.1 Pi:

   echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-0/new_device

   Type exit to stop being the 'root' user.

2. Now check the time on the RTC device using:

   sudo hwclock –r

   If this is the first use of the RTC, it will report back Jan 1 2000.

   Please configure now the current time and then conform with:

   sudo hwclock –w

   to write the system time to the RTC module.

3. To set up the Raspberry Pi using the Real Time Clock, you will need to add the RTC module chip to the file ***/etc/modules.***

   - Go into the terminal window and run:

sudo nano /etc/modules

Then add ***rtc-ds1307*** at the end of the file.

- Next step is to add the DS1307 device creation at boot by editing the file ***/etc/rc.local***

  Run ***sudo nano /etc/rc.local***

  and add the following lines to the end of the file:

  echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device

  sudo hwclock -s

  date

  To save the file, press Ctrl+X, Y then return.

4. The next time you reboot your Raspberry Pi it will read the current time from the Real Time Clock on the S.USV.

4 Client Software

Communication between S.USV and the Raspberry Pi via the I2C interface at the address 0x0F.
In principle the software package of the S.USV consist of two tools:

1. The S.USV Daemon (susvd), which monitors and controls the S.USV by constantly reading the S.USV status and reacting on several events. The S.USV Daemon will be started once and is running in the background.

2. The S.USV Client (susv), which gives the user the possibility to get and see the actual status of the S.USV as well as to control the S.USV, e.g. activating or deactivating the charging circuit. The S.USV-Client is also responsible for editing the config-variables of the S.USV Daemon, eg. Shutdown timer.

4.1 S.USV – Daemon

The S.USV daemon is responsible for monitoring and controlling the S.USV in conjunction with the Raspberry Pi.

The S.USV daemon creates a log in the file: ***/var/log/susvd.log***

The following sections will be shown the individual options.

4.1.1 Daemon Configuration

To configure the S.USV Daemon following commands are available:

(Please use this commands with root privileges)

sudo ./susv -timer <time in seconds>

This value indicates how long the system continues to run before the filesafe shutdown will be initiated by the S.USV after the voltage supply is switched to battery.

Values “>=0” are possible.

This function can be disabled by the value “-1”.

In this case, the Raspberry Pi remains running, until the Battery Capacity reaches 10%. From that point on, the S.USV will automatically perform a filesafe shutdown in order to prevent the battery from getting damaged.

sudo ./susv -auto <0/1>

This value determines the starting behavior of the S.USV Daemon. The value "1" activates the autostart, the value "0" disables the autostart.

If the autostart is disabled, please notice that you have to start the S.USV-Daemon manually in order for the S.USV to work correctly.

sudo ./susv -sleep <time in seconds>

This value determines the repetition in which the S.USV Daemon monitors and controls the voltage output of the S.USV. Values “>=0” are possible.

4.1.2 Daemon Controlling

To control the S.USV Daemon following commands are available:

(Please use this commands with root privileges)

sudo ./susvd -start

Starts the S.USV Daemon Service and its configuration.

sudo ./susvd -stop

Stops the S.USV Daemon Service.

sudo ./susvd -restart

Restarts the S.USV Daemon Service.

4.2 S.USV – Client

The S.USV Client allows the user state monitoring and function control of the S.USV.

The following sections will be shown the individual options.

4.2.1 Client Options

To control the S.USV Client following commands are available:

./susv -help

This command shows all possible options.

./susv -chrgon

Switch charging circuit on.

Allows the manual switching of the battery's charging circuit.

./susv -chrgoff

Switch charging circuit off.

Allows the manual shutdown of the battery's charging circuit.

./susv -vin

Read input voltage.

With this command it is possible to check the current input voltage.

./susv -pwrext

Read the external power consumption.

With supply via the external voltage input, the current external power consumption can be checked by this command.

./susv -pwrbat

Read the battery power consumption.

With supply via battery, the current battery power consumption can be checked by this command.

./susv -capbat

Read battery capacity.

This command allows to read the current battery voltage and the remaining battery capacity.

(Note: A remaining capacity of <25% is indicated by the LED BATRD. At a remaining capacity of <10% the Raspberry Pi will automatically shut down.)

sudo ./susv -chrgpwr <300/500/1000>

(Please use this command with root privileges)

Change the charge current for the battery.

Use this command to change the active charge current for the battery. In order to minimize the charging time following current strengths are available.

- 300mA
- 500mA
- 1000mA

(Note: The configured charging current is stored in EEPROM and reloaded at startup.)

sudo ./susv –chgadd <0x..>

(Please use this command with root privileges)

Change the I2C address of the S.USV.

To avoid potential compatibility issues, use this command to change the I2C address of the S.USV. Verify the address by typing ***i2cdetect –y 1*** in the terminal.

(Note: The configured I2C address is stored in EEPROM and reloaded at startup.)

./susv –flash <path to HEX file>

Upgrade the firmware.

Use this command to upgrade the actual firmware.

./susv -status

Read S.USV status.

This command allows to read the S.USV status. Here of all available modes are indicated as well as the current powering source and its power consumption.

Seprotronic GmbH / Helmsheimer Straße 28 / 70499 Stuttgart
Tel. +49 (0) 7 11 - 2 20 70 79 - 0 / Fax +49 (0) 7 11 - 2 20 70 79 - 9
E-Mail Info@s-usv.de / www.s-usv.de